Ответы на вопросы домашнего задания № 1.

Перечень вопросов:

1. Как организовать клиент-серверное взаимодействие?
2. Как и в каком виде передавать файлы?
3. Как пересылать большие файлы?
4. Как пересылать служебные команды?
5. Что хранить в базе данных?
6. Как передавать структуру каталогов/файлов?
7. Организация клиент-серверное взаимодействия

Взаимодействие между сервером и клиентом построить по принципу запрос ответ. За формирование запросов и логики взаимодействия будет отвечать ПО клиента.

Сообщения между сервером и клиентом имеют следующий формат:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Синхро- последовательность | | | | Номер посылки | Длина посылки | Тело посылки | КС |  |
| …. | 0xF0 | 0xE7 | 0xC3 | 0x81 | SHORT | SHORT | N - байт | SHORT | …. |

**Синхропоследовательность** – это последовательность байт, необходимая для выделения начала посылки. Если в данных встречается последовательность байт совпадающая с синхропоследовательностью, то после этих байт делается вставка из инверсной синхропоследовательности (0x0F, 0x18, 0x3C, 0x7E), которая указывает что данные байты не являются синхропоследовательностью. На приёмной стороне данная вставка удаляется из данных.

**Номер посылки** – счётчик посылок с автоинкрементном и автосбросом в ноль при переходе через максимум, со значениями от 0 до 32767.

**Длинна посылки** – длинна тела посылки в байтах 0 до 32767;

**КС – контрольная сумма** – сумматор байт номера посылки, длинны посылки и тела посылки (так как в Java нет без знаковых чисел, то КС рассчитывается как сумма знаковых байтовых чисел).

Длина посылки и КС рассчитываются с учётом вставок инверсных синхропоследовательностей.

Тело посылки в свою очередь содержит код команды или операции, параметры и данные.

После получения посылки, принимающая сторона формирует квитанцию об успешном или не успешном её получении, следующего вида:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Синхро- последовательность | | | | Номер посылки | Длина посылки | Инверсия КС |  |
| …. | 0xF0 | 0xE7 | 0xC3 | 0x81 | SHORT | SHORT | SHORT | …. |

Получение квитанции так же говорит о том, что можно высылать новое сообщение или повторить ранее переданное.

1. Передача файлов

Файлы передаются в виде байтового потока, нарезанного на кадры, заданного размера. Последний кадр, передающий остаточные данные, может быть короче. При передаче файла в теле посылки указывается следующая информация:

* код команды «передача файла»;
* номер сеанса передачи файла (для каждого последующего файла он будет новым);
* количество кадров, на которые разбит файл;
* номер кадра файла;
* количество данных в кадре (примечание: данное количество указывается в байтах без учёта возможных вставок инверсных синхропоследовательностей);
* данные файла.

В первом кадре в место данных передаётся мето-данные файла (в виде сток) : «имя», «путь», «размер», «дата создания», «дата сохранения».

Сервер после приёма первого кадра формирует файл с временным именем, допустим полученным как соединение имени хранилища и номера сеанса передачи данных. Далее с получением каждого последующего кадра, полученные данные добавляются к временному файлу. И это происходит до того момента, как не будет получен последний кадр. Затем файл регистрируется в БД, ему присваивается новое обезличенное имя, с которым в базу данных вносятся полученные мето-данные файла.

Для контроля процесса прохождения и заверения передачи файла необходимо предусмотреть команды запроса статуса процесса пересылки файла. Это необходимо для завершения процесса пересылки при потере соединения. В ответ на запрос, приходит либо ответ с кодом, говорящим о том, что незавершённого процесса передачи нет, или же код ошибки с указанием мета-данных файла, номера сеанса передачи файла и количеством принятых данных (в байтах). По принятым данным формируются посылки завершающие процесс передачи файла.

1. Передача служебных команд

Служебные команды передаются схожим образом. В теле посылки передаётся код команды, затем указываются параметры. В отличие от пересылки файлов, служебные команды меньше по размеру и должны помещаться в одну посылку.

1. Построение базы данных

Хранение данных на сервере должно быть реализовано в виде база данных (БД), которая содержит следующие связанные таблицы:

1. repositories – хранилища;
2. directories – директории (папки);
3. files – файлы;
4. users – пользователи;
5. admin\_log – журнал администратора.

Файлы данных хранятся на диске сервера в обезличенном виде, под именами сформированными БД автоматическим способом.

В БД в таблице files храниться:

* ссылка на новое обезличенное имя файла;
* идентификатор хранилища;
* идентификатор папки в таблице directories;
* оригинальное имя файла на источнике;
* путь, по которому файл находится на источнике;
* размер файла;
* дата создания файла (возможно);
* дата последнего сохранения файла.

В БД в таблице repositories храниться:

* идентификатор хранилища;
* имя хранилища;
* полный путь к папке хранилища на сервере;
* идентификатор пользователя – «HOST» клиента;
* объём занятого пространства в хранилище (в байтах);
* размер хранилища (в байтах).

В БД в таблице directories храниться:

* идентификатор папки;
* имя хранилища;
* идентификатор внешней папки;
* имя папки в исходнике;
* полный путь к папке на источнике.

Примечание – само хранилище является папкой (корневой), которая также прописывается в БД в таблице directories для того, чтобы папки хранилища могли ссылаться на неё как на внешнюю папку. В качестве параметра <идентификатор внешней папки> для неё указывается значение NULL.

В БД в таблице users храниться:

* идентификатор пользователя;
* логин пользователя;
* пароль пользователя.

В БД в таблице admin\_log храниться:

* идентификатор записи в журнале;
* идентификатор пользователя;
* идентификатор папки (для «HOST» клиента указывается идентификатор корневой папки хранилища);
* флаг разрешения чтения файлов в папке
* флаг разрешения записи файлов в папку
* флаг разрешения перезаписи файлов в папке
* флаг разрешения переименования файлов в папке
* флаг разрешения удаления файлов папки
* флаг разрешения создания папок
* флаг разрешения переименования папок
* флаг разрешения удаления папок

1. Передача структуры каталогов/файлов

Для организации процесса синхронизации данных между исходником и хранилищем, а также индикации у клиентов структуры папок и списков файлов хранилища, наиболее рациональным будет формирование одной посылки от хранилища к клиенту содержащей всю необходимую информацию, включая мета данные файлов.

Для этого предполагается использовать формат данных JSON.